

CLAIMS

What is claimed is:

1 1. A computer data signal embodied in a carrier wave and representing
2 sequences of instructions which, when executed by a processor, causes the processor
3 to provide a separate Java Virtual Machine to instantiate a thread in a data processing
4 system, such that the processor:

5 receives at least one request for service;

6 identifies a resource requested and a class corresponding to the resource;

7 if a virtual machine is available, reuses the available virtual machine by a
8 single thread only by loading the virtual machine to instantiate the thread
9 corresponding to the requested resource, wherein the virtual machine is available if
10 the virtual machine has a matching process identification and thread identification and
11 is not busy; and

12 if the virtual machine is unavailable, spawns and loads a new virtual machine
13 to instantiate the thread corresponding to the requested resource, wherein the virtual
14 machine is unavailable if either the virtual machine does not have the matching
15 process identification and thread identification or is busy.

1 2. The computer data signal of claim 1, wherein the processor further
2 receives the request for service by a web client.

1 3. The computer data signal of claim 1, wherein identifying the resource
2 requested comprises:

3 parsing a Uniform Resource Locator; and

4 using the Uniform Resource Locator to identify the class that hosts a program
5 that when executed provides the resource requested.

1 4. The computer data signal of claim 1, wherein the processor further
2 enqueues the resource request before determining if the virtual machine is available.

1 5. The computer data signal of claim 1, wherein the processor further:
2 dequeues the resource request after determining if the virtual machine is
3 available;
4 routes the resource request to the virtual machine where the virtual machine is
5 available; and
6 routes the resource request to the new virtual machine where the virtual
7 machine is unavailable.

1 6. The computer data signal of claim 3, wherein the program is a servlet.

1 7. The computer data signal of claim 1, wherein the processor sends an
2 output stream to an origin of the resource request.

1 8. A computer program product comprising:
2 a computer usable medium having computer readable code embodied therein
3 for causing a unique Java Virtual Machine to instantiate a thread in a data processing
4 system, the computer program product comprising:

5 a first computer readable program code configured to cause a computer
6 to effect receiving at least one request for service; and

7 the first computer readable program code configured to cause the
8 computer to effect identifying a resource requested and a class corresponding
9 to the resource;

10 the first computer readable program code configured to cause the
11 computer to effect if a virtual machine is available, reusing the available
12 virtual machine by a single thread only by loading the virtual machine to
13 instantiate the thread corresponding to the requested resource, wherein the
14 virtual machine is available if the virtual machine has a matching process
15 identification and thread identification and is not busy; and

16 the first computer readable program code configured to cause the
17 computer to effect if the virtual machine is unavailable, spawning and loading
18 a new virtual machine to instantiate the thread corresponding to the requested
19 resource, wherein the virtual machine is unavailable if either the virtual
20 machine does not have the matching process identification and thread
21 identification or is busy.

1 9. The computer program of claim 8, wherein the first computer readable
2 program code is configured to cause the computer to effect sending a service request
3 by a web client.

1 10. The computer program of claim 8, wherein identifying the resource
2 requested comprises:
3 parsing a Uniform Resource Locator; and
4 using the Uniform Resource Locator to identify the class that hosts a program
5 that when executed provides the resource requested.

1 11. The computer program of claim 8, wherein the first computer readable
2 program code is configured to cause the computer to effect enqueueing the resource
3 request before determining if the virtual machine is available.

1 12. The computer program of claim 8, further comprising:
2 the first computer readable program code configured to cause the computer to
3 effect dequeuing the resource request after determining if the virtual machine is
4 available;
5 the first computer readable program code configured to cause the computer to
6 effect routing the resource request to the virtual machine where the virtual machine is
7 available; and
8 the first computer readable program code configured to cause the computer to
9 effect routing the resource request to the new virtual machine where the virtual
10 machine is unavailable.

1 13. The computer program of claim 10, wherein the first computer
2 readable program code is a servlet.

1 14. The computer program of claim 8, wherein the first computer readable
2 program code is configured to cause the computer to effect sending an output stream
3 to an origin of the resource request.

1 15. A method performed in a data processing system, by at least one
2 processor, comprising:

3 receiving a plurality of requests for service from a client;

4 identifying a respective servlet for each of the plurality of requests and a class
5 corresponding to each of the respective servlets; and

6 providing a separate virtual machine for each of the respective servlets, each
7 separate virtual machine executing in a separate addressable space, wherein if a
8 virtual machine is available then reusing the available virtual machine by a single
9 thread by loading the virtual machine to instantiate a thread corresponding to the
10 requested service, and wherein the virtual machine is available if the virtual machine
11 has a matching process identification and thread identification and is not busy, and

12 if no virtual machine is available then spawning and loading a new virtual
13 machine to instantiate the thread corresponding to the requested service, and wherein
14 no virtual machine is available if either the virtual machine does not have the
15 matching process identification and thread identification or is busy.

1 16. The method of claim 15, wherein the receiving further comprises
2 receiving the plurality of requests for service from at least one web client.

1 17. The method of claim 15, wherein the receiving further comprises
2 identifying a resource requested.

1 18. The method of claim 15, wherein the receiving the resource requested
2 further comprises the steps of:

3 parsing a Uniform Resource Locator; and

4 using the Uniform Resource Locator to identify the class that hosts a program
5 that when executed provides the resource requested.

1 19. The method of claim 15, further includes the step of:

2 enqueueing a resource request before determining if the virtual machine is
3 available.

1 20. The method of claim 15, further includes the step of:
2 dequeuing a resource request after determining if the virtual machine is
3 available;
4 routing the resource request to the virtual machine where the virtual machine
5 is available; and
6 routing the resource request to the new virtual machine where the virtual
7 machine is unavailable.

1 21. The method of claim 18, wherein the servlet is the program.

1 22. The method of claim 15, further comprising the step of sending an
2 output stream to an origin of the resource request.